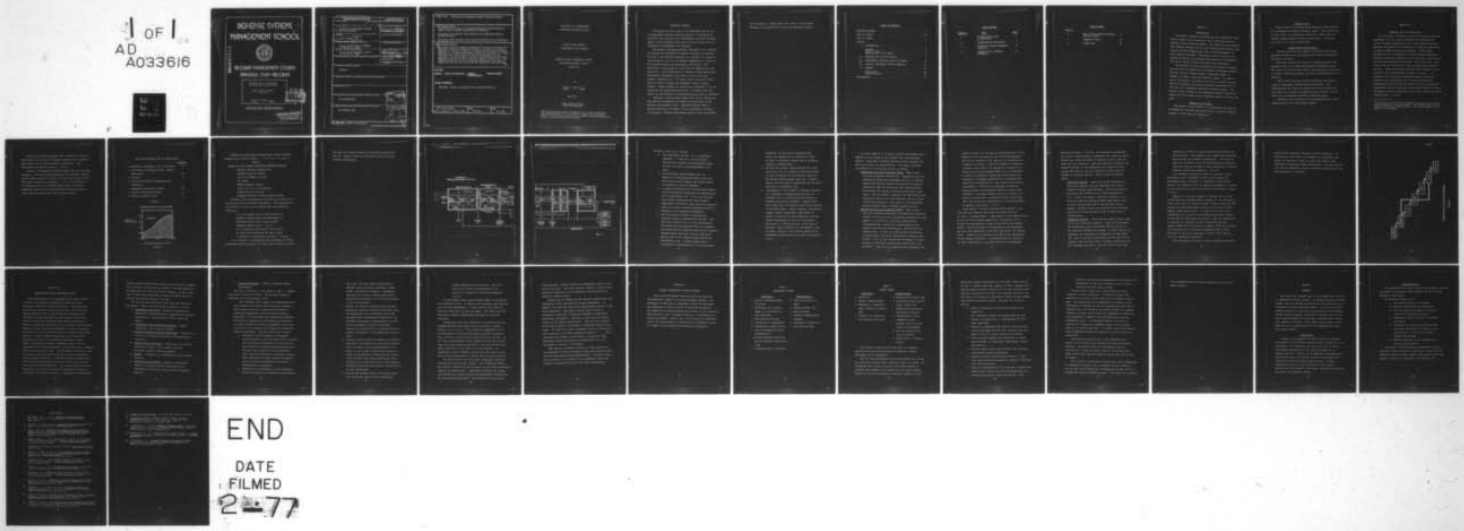
AD-A033 616 DEFENSE SYSTEMS MANAGEMENT SCHOOL FORT BELVOIR VA F/G 9/2
THE ROLE OF AN INDEPENDENT SOFTWARE VALIDATION AGENCY.(U)
MAY 76 C R MAGILL

UNCLASSIFIED NL

1 OF 1
AD
A033616



END

DATE
FILMED
2-77

DEFENSE SYSTEMS MANAGEMENT SCHOOL

PROGRAM MANAGEMENT COURSE
INDIVIDUAL STUDY PROGRAM

THE ROLE OF AN INDEPENDENT
SOFTWARE VALIDATION AGENCY

STUDY PROJECT REPORT
PMC 76-1

Clyde R. Magill, Jr.
LTCOL USAF

FORT BELVOIR, VIRGINIA 22060

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)*: THE ROLE OF AN INDEPENDENT SOFTWARE VALIDATION AGENCY.	5. TYPE OF REPORT & PERIOD COVERED: Study Project Report 76-1
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Clyde R. Magill, Jr	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Defense Systems Management ~~College~~ Sch, Ft. Belvoir, VA 22060	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Defense Systems Management College, Ft. Belvoir, VA 22060	12. REPORT DATE: ~~1975~~ May 76 13. NUMBER OF PAGES: 39 (12) 40p.
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*	15. SECURITY CLASS. *(of this report)*: UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

UNLIMITED

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

SEE ATTACHED SHEET

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

SEE ATTACHED SHEET

ACCESSION for
NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

408 462

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

STUDY TITLE: The Role of an Independent Software Validation Agency.

STUDY PROJECT GOALS:

To investigate the role of an Independent Software Validation Agency in order to improve computer program test and evaluation.

To determine the essential tasks required of an Independent Software Validation Agency.

To provide a more realistic set of tasks for consideration by independent test agencies.

STUDY REPORT ABSTRACT:

New systems containing software development will continue to increase the Department of Defense investment in software. The cost of the software programs will continue to increase in relationship to the cost of hardware components of a system as more complex systems are developed in the future. This importance is due primarily to the cost of software test and evaluation. The testing phase of computer program design and development represents 50 percent or more of the total computer software costs. This report evaluates the role of an independent test agency and provides an updated and realistic set of tasks for consideration by these agencies.

KEY WORDS

MATERIEL **DESIGN AND DEVELOPMENT** **COMPUTERS** **COMPUTER SOFTWARE**
SYSTEMS ANALYSIS

SYSTEMS ENGINEERING

KEY WORDS: Software, Independent Test, Test and Evaluation.

NAME, RANK, SERVICE	CLASS	DATE
Clyde R. Magill, Jr., LTCOL, USAF	PMC 76-1	May 1976

THE ROLE OF AN INDEPENDENT SOFTWARE VALIDATION AGENCY

Study Project Report

Individual Study Program

Defense Systems Management School

Program Management Course

Class 76-1

by

Clyde R. Magill, Jr.
LTCOL USAF

May 1976

Study Project Advisor
LCDR Sue Anderson

EXECUTIVE SUMMARY

The purpose of this study is to investigate the role of an Independent Software Validation Agency, to determine the essential tasks required of an Independent Software Validation Agency, and to provide a more realistic set of tasks for consideration by independent test agencies.

New systems containing software development will continue to increase the Department of Defense investment in software. The cost of the software programs will continue to increase in relationship to the cost of hardware components of a system as more complex systems are developed in the future. This importance is due primarily to the cost of software test and evaluation. The testing phase of computer program design and development represents 50 per cent or more of the total computer software costs. To reduce these costs every effort should be made to reduce the redundance found in system testing. Recent emphasis on requiring an independent test and evaluation has added additional costs. To reduce this cost impact new and innovative testing philosophies must be developed.

"Cognizant" testing would reduce this cost by getting the user and the independent test agency involved early in the software developmental cycle. Combined DT&E and OT&E of software decreases the number of test conducted to validate the software. Software development requires early involvement

by all agencies. Waiting until the system is "delivered" increases the potential for costly and redundant testing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SECTION I

INTRODUCTION

The author selected the subject for this individual study report because of the experience that he has obtained while working on major defense programs. The author has been dealing with computer programs for the past fourteen years and has seen some of the major changes take place in the development of major computer programs. The author elected to look at the current directives and available documentation to determine if additional insight could be provided by his experience.

The documents reviewed indicated that extensive progress has been made in managing the development of computer program portions of major weapon systems. These data could, in themself, develop a sound and important report. In order to tailor this report, the report will focus on computer program test and evaluation and provide important recommendations on the role of an Independent Software Validation Agency. The specific tasks assigned to an Independent Software Validation Agency have just began to be an important issue in the past few years.

Purpose of the Study

The purpose of this report is to investigate the role of an Independent Software Validation Agency in order to improve computer program test and evaluation.

Specific Goal

The aim will be to determine the essential tasks required of an Independent Software Validation Agency. The application of these tasks to a particular program will assure that the computer program meets the system requirements. The report will focus on Air Force acquisitions.

Organization of the Report

The next section of the report will review the software acquisition process and will focus on the test and evaluation portion of the cycle.

The third section will look at a software contract that assigned tasks to an Independent Software Validation Agency. The review will relate these tasks to the acquisition cycle and to several problems associated with the implementation of this contract.

Section four will deal with the advantages provided by using an Independent Software Validation Agency. The disadvantages that might be incurred will also be discussed. The author will provide an updated and more realistic set of tasks for an Independent Software Validation Agency.

The report will conclude with recommendations for future acquisitions of Air Force weapon systems.

SECTION II

SOFTWARE TEST AND EVALUATION

The importance of taking a look at the computer software development cycle is just as important as looking at the typical hardware acquisition cycle. New systems containing software development will continue to increase the Department of Defense (DOD) investment in software. "Total annual expenditures for system analysis, design and programming of software in DOD are estimated at $3-3.5 billion, divided among the Military Departments as follows: Army 23 per cent, Navy 36 per cent, Air Force 36 per cent, and other DOD agencies 5 per cent. Other studies have provided some estimates of the software cost by application, as shown in Figure 1. If management and logistic information systems are taken as primarily data processing, and if aircraft and missile engineering and production are taken as primarily scientific programming, then the remainder, which might loosely be called the weapon systems, constitutes 55 to 75 per cent of the total software cost." (7:29)[1]

---

[1] This notation will be used throughout the report for sources of quotations and major references. The first number is the source listed in the bibliography. The second number is the page in the reference.

The cost of software programs will continue to increase in relationship to the cost of hardware components of a system as more complex systems are developed in the future. This relationship is shown in Figure 2. (9:24)

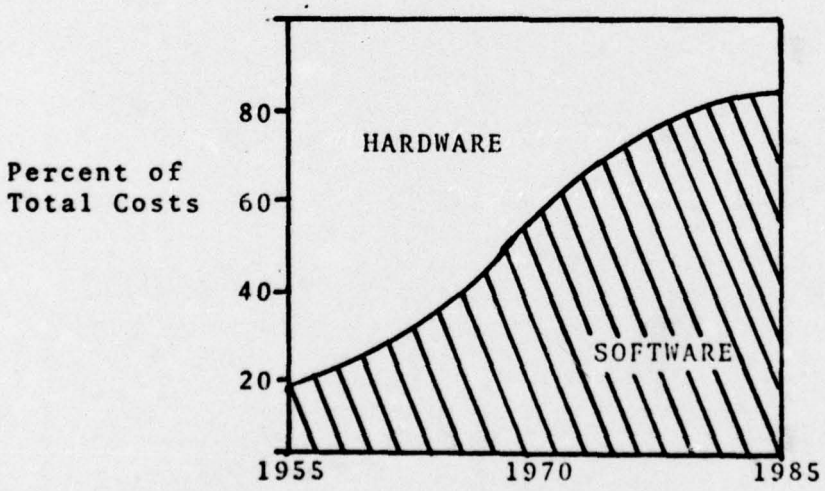
Figure 2 is supported by another author with the following statement: "Current annual expenditures for embedded computer systems (ECS) exceed $2 billion, with more than 70 per cent of this amount dedicated to software." (4:2) Since there is an important part of our defense budget spent on software, management should be very concerned about developing ways to insure maximum benefits from the tax payers money.

AIR FORCE SOFTWARE COST BY APPLICATIONS

	Percent
o Research, Development, Test & Evaluation	28
o Intelligence and Communication, Command and Control	19
o Avionics	9
o Aircraft and Missile Engineering and Production	5
o Management Information Systems	14
o Logistic Information Systems	5
o Other and Indirect Costs	20

Figure 1

HARDWARE/SOFTWARE RELATIONSHIP



Hardware/Software Trends

Figure 2

Examples of major weapon systems which contain computer programs are listed in Table 1. (4:2, 6:iv, 3:4, 9:23)

TABLE 1

MAJOR AIR FORCE SYSTEMS CONTAINING COMPUTER PROGRAMS

Advanced Airborne Command Post

Advanced Logistics System

A-10 Close Support Aircraft

B-1 Bomber

Defense Support Program

F-15, F-16, F-17, F-18 Fighters

Minuteman Missile System

Safeguard Ballistic Missile Defense System

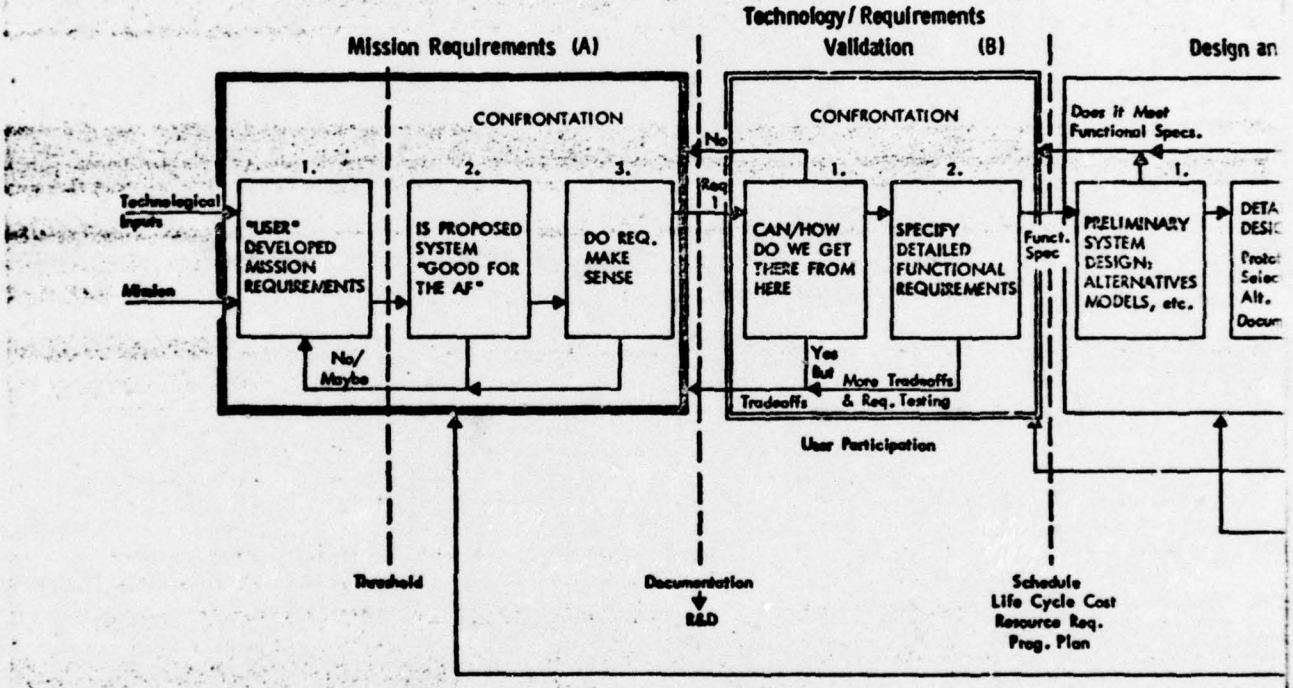
Software test and evaluation is the key part of the total acquisition cycle of software development. This importance is due primarily to the cost of software test and evaluation. "Typically:

> 'It is not unusual for the testing phase of computer program design and development to represent 50 per cent or more of total computer program costs.' (11:3)
>
> 'It is recognized that close to 50 per cent or more of the production of large systems is devoted to the period of testing.' (13:123)

It is necessary to concentrate upon performing the entire development process properly the first time in order to attain

the goal of a quality product at the lowest dollar cost." (8:1-1) Figure 3 shows the acquisition cycle of typical software developments.

ILLUSTRATIVE
"SYSTEM DEVELOPMENT" LIFE CYCLE



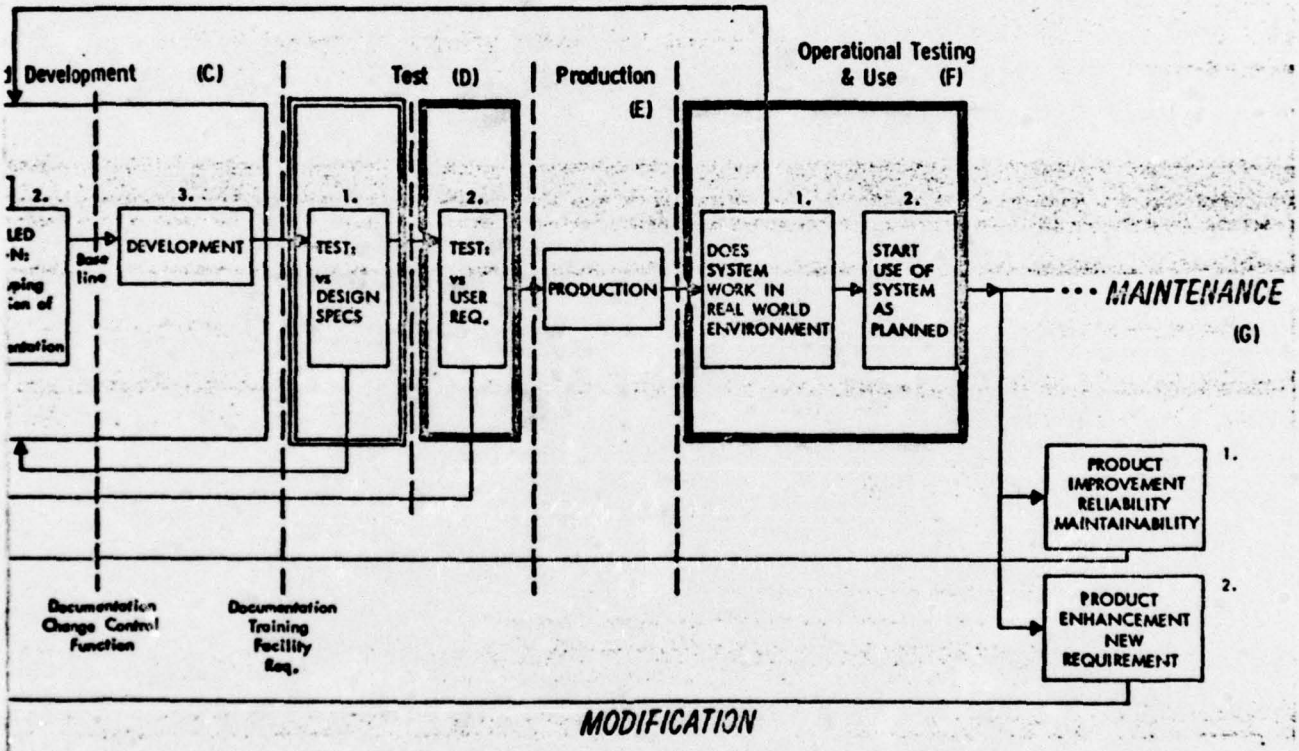

Development (C)
Test (D)
Production (E)
Operational Testing & Use (F)
2.
3.
1.
2.
1.
2.
DEVELOPMENT
Base line
TEST: vs DESIGN SPECS
TEST: vs USER REQ.
PRODUCTION
DOES SYSTEM WORK IN REAL WORLD ENVIRONMENT
START USE OF SYSTEM AS PLANNED
MAINTENANCE (G)
PRODUCT IMPROVEMENT RELIABILITY MAINTAINABILITY
1.
PRODUCT ENHANCEMENT NEW REQUIREMENT
2.
Documentation Change Control Function
Documentation Training Facility Req.
MODIFICATION

Fig. 3

The major areas are as follows:

"1. The requirements process (A) is exceedingly important. It must be a user-controlled function that provides the performance inputs to drive the entire system-development life cycle.

2. The functional system designer (B), in addition to translating the needs of the user, must interact with computer and system experts to identify and analyze tradeoffs.

3. A careful distinction must be maintained between various requirements documents and implementation of the requirements (C). It is to be expected that several iterations will occur between requirements and design; thus, each will gradually evolve to a final position reflecting technical feasibility and acceptable performance.

4. The line coding of Figure 3 indicates the different kinds of organizations involved. The bold-line functions (A.D.F.) are user-oriented; the double-line test function (D) is an independent group and the validation function (B) involves computer-system expertise; and the light-line function (C) is the computer-system design and development group. A single command could conceivably be responsible for all the functions

indicated, but the various "organizations" within the command must be distinct in order to react to different subgoals and to encourage productive confrontation.

5. The user must remain involved during the entire life cycle, but in a carefully controlled manner to avoid intruding into the orderly conduct of the design-and-development phase (C). Significantly, the user must be responsible for the final test prior to production (D).

6. 'Maintenance' in the context of computer software is a thoroughly misleading word and does not correlate with its well-established meaning in hardware and logistics matters. Computer software simply does not break and have to be repaired in the usual sense. Rather, software 'maintenance' expands product improvement, improvement in product reliability or maintainability, and even product enhancement to meet new requirements. This point is stressed because, in the case of software, these attributes of 'maintenance' tend to force a return to much earlier phases of the development process than in the case of hardware." (6:6)

By using Figure 3 as a typical software development cycle, emphasis can be placed on the software test and evaluation process. Department of Defense Directive 5000.3 provides the basic policy for test and evaluation. Two types of testing are defined in the directive and are as follows:

Development Test and Evaluation (DT&E). DT&E is that test and evaluation conducted to: demonstrate that the engineering design and development process is complete; demonstrate that the design risks have been minimized; demonstrate that the system will meet specifications; and estimate the system's military utility when introduced. DT&E is planned, conducted, and monitored by the developing agency of the DOD Component, and the results thereof are reported by that agency to the responsible Military Service Chief or Defense Agency Director.

Operational Test and Evaluation (OT&E). OT&E is that test and evaluation conducted to estimate the prospective system's military utility, operational effectiveness, and operational suitability (including compatibility, interoperability, reliability, maintainability, and logistic and training requirements), and need for any modifications. In addition, OT&E provides information on organization, personnel requirements, doctrine, and tactics. Also, it may provide data to support or verify material in operating instructions, publications, and handbooks. OT&E will be accomplished by operational and

support personnel of the type and qualifications of those expected to use and maintain the system when deployed, and will be conducted in as realistic an operational environment as possible. OT&E will normally be conducted in phases, each keyed to an appropriate decision point. During Full-Scale Development OT&E will be accomplished to assist in evaluating operational effectiveness and suitability (including compatibility, interoperability, reliability, maintainability, and logistic and training requirements). OT&E will be continued as necessary during and after the production period to refine these estimates, to evaluate changes, and to re-evaluate the system to insure that it continues to meet operational needs and retains its effectiveness in a new environment or against a new threat. (5:2)

Using Figure 3 it can be seen that DT&E relates to the test that are conducted (D) within the double-line box (1. Test: US Design Specs). DT&E usually is the responsibility of the Air Force Systems Command and is conducted jointly between the developmental contractor and the System Program Office. The application of the independent test philosophy has been used effectively by software contractors for several years. "Errors in the software exist due to the subjectivity of the software developer. An accepted means of correcting for this subjectivity is the application of an independent

testing philosophy." (6:5-10) The contractor accomplished this task by establishing an independent test group and transferring the software developer's programs to this group for final test and evaluation. When the contractor transfers the software programs to the independent test group he either assigns the group to test the software by using black box testing or cognizant testing. Each of these are defined below:

"Black Box Testing" - where the testing performed by a test group separate from the developers and without complete software specifications. The testing group is only given the software and the test specifications (i.e., range of variables, accuracies required, etc.). They are given any testing sequence requirements known to exist and are given user's instructions with I/O formats. Detected errors are reported to the developer group and no attempt is made by the testing group to correct errors.

"Cognizant Testing" - where the test group is fully aware of how the software has evolved. Using this philosophy the independent tester understands the how and why of the selected implementation method. An early decision to implement this philosophy of independent testing means that there is an early understanding on the part of the separate group and that there is greater opportunity for the interchange of ideas. Its use implies that a good

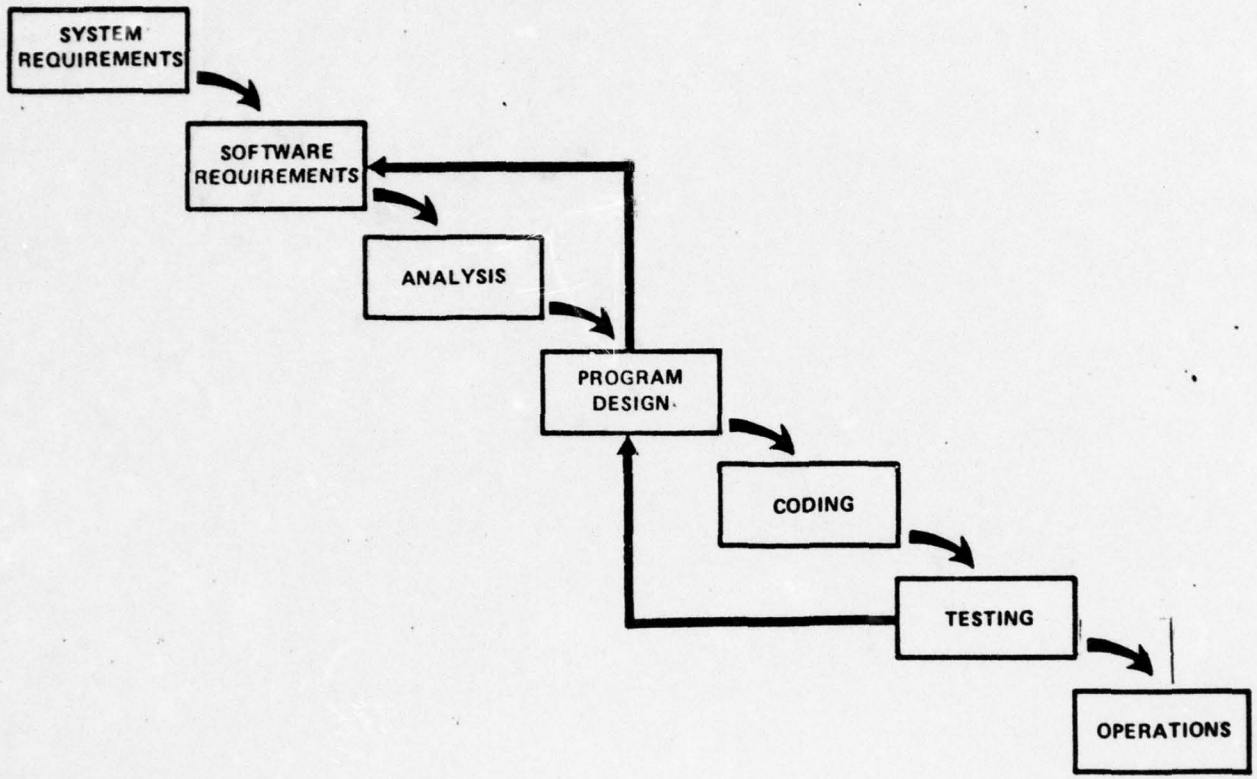
communication link will exist between the developing and testing groups. This approach will require heavy emphasis upon accurate and complete documentation. This type of group can be expert in the use of automated test measurement tools and can use them to great advantage in insuring complete testing for acceptance. (6:5-10)

If a contractor decides on using the cognizant testing philosophy he must implement it early in the design and development phase (C) Figure 3. Regardless of the choice, however, the iterative cycle of software development is focused on the outcome of the tests conducted on the software. Figure 4 (12:3).

Again, by referring to Figure 3, the Test (D) (2. Test: vs User Req) and the Operational Testing & Use (F) outlined by solid black boxes are related to OT&E. Prior to 1974 the OT&E testing was conducted jointly between the Air Force Systems Command (AFSC) and the user. In early 1974 the Air Force established the Air Force Test and Evaluation Center (AFTEC) at Kirtland Air Force Base, New Mexico. This new organization is separate and distinct from the developing/procuring command (AFSC) and from the using command. AFTEC was assigned the responsibility of providing independent OT&E ad uses AFR 80=14 as the basic regulation to govern their testing. AFR 80-14 implements DOD 5000.3.

This section has outlined a typical software development

cycle and has focused on software test and evaluation. An introduction of DOD 5000.3 and AFTEC as an independent test agency was provided in order to relate the author's past experience to potential future requirements. The next section will look at Independent Software Validation Agency and at the tasks assigned to the agency.



Iterative Cycle of Software Development

Figure 4

SECTION III

INDEPENDENT SOFTWARE VALIDATION AGENCY

The establishment of an independent test agency (AFTEC) in the Air Force might be construed by some as a new an innovative approach to system acquisition. The author does not entirely agree with this point of view. The Air Force Systems Command (AFSC) for years has been concerned about developing systems that meet the requirements of the users. AFSC has always been concerned about applying resources to obtain an independent assessment prior to major system decisions. Example in point are system acquisitions in which consultants are obtained to provide outside advice on specific acquisition problems. Another example might be the use of non-profit organizations such as the Aerospace Corporation. These types of corporations provide independent analysis of the requirements for system development and the major tests conducted during the acquisition cycle. Another example the author would like to deal with specifically was the use of an independent system software contractor. The Defense Support Program (DSP) used the System Development Corporation (SDC) as an independent system software contractor. The primary function of this contractor was in fact to provide the independent evaluation called for in DOD directive 5000.3. The degree of independence

from the System Program Office (SPO) would be the only argument that it did not in fact meet the intent of the DOD directive. The important point that can be made, however, is that it did focus on the basic requirements of getting another agency to validate the software design of the DSP.

The contract had seven basic areas that were tasked on this agency. These tasks are listed as follows: (15:26)

1. Development Monitoring. Primarily concerned with monitoring software contractor compliance with system requirements. Participation in PDR's, CDR's and FACI's.
2. Independent System Validation Testing. Conduct independent test on system software.
3. Software Interface Baseline Maintenance. Primarily to control all software/software and software/hardware interfaces.
4. Program Library Operation. Single point to maintain developers software, user software and other government agencies software programs.
5. Support. Primarily for supporting the user at field locations.
6. Configuration Management. Maintain configuration control of aggregate programs generated by the developmental contractor, user and other government agencies.

7. System Improvements. Identify suggested system improvements.

The task of interest in this report is task 2 - Independent System Validation Testing. The specific tasking is extracted from the statement of work.

> The contractor shall support the integrated system test program at the Sensor Data Processing Laboratory, Multi Purpose Facility, or operating locations by conducting independent tests on selected software subsystems. The Contractor shall develop and maintain the materials required for the testing and validation of system releases and conduct independent testing and validation of the software in accordance with approved plans and procedures. Specific tasks shall include:
>
> a. Preparation and maintenance of independent system validation test plans which shall establish the verification requirements and criteria for the validation of the data processing subsystem. The plan shall specify the test environment, tools and the necessity for incremental testing where applicable and shall provide traceability with the system requirements and interface requirements as appropriate.
>
> b. Preparation and maintenance of the independent system validation test procedures which describe

the tests. The test shall be described in detailed terms specifying objectives, inputs, events, and expected responses. Information regarding the schedule, manning requirements, and operating procedure shall also be contained in the test procedures.

c. Conduct the required software system and data processing interface testing in accordance with the independent system validation test plans and procedures to demonstrate capabilities and interface requirements and participate in additional operational testing to the extent necessary to insure that the software system will operate successfully in the operational environment.

d. Support release control by rendering an objective assessment on the quality of the tested software. This task shall include identification of the effect on the overall system of the release in terms of the software's capabilities and limitations stated in functional, operational terms, as well as the characterization of discrepancies, deficiencies and anomalies against specifications or test requirements.

e. Prepare and maintain system test reports which will reveal the results of the independent

system validation test activities. This task shall also include the maintenance of the results of correction of errors/discrepancies. (15:28)

At first glance these specific tasks might be interpreted as an exhaustive list. In spite of this, problems stem from how the tasks are implemented. In this contract the contractor used the "black box testing" philosophy. The "black box" was the entire software program being developed by the prime contractor.

Independent tests were conducted at the system level and accomplished after each major build of the basic software (System 5.2, 5.3, etc.). This caused the independent tester to invariably be several builds behind the prime contractor. This, of course, was an area of major concern to the SPO. By the time the independent tester provided the SPO with problem areas the prime contractor had built a new baseline and would always indicate he had solved all of the problems. The independent tester, therefore, had to drop all actions on the last test and concentrated on testing the new system baseline.

The primary driver for this early independent testing of evolving baselines was the schedule. The independent testing was formally scheduled at the completion of the prime contractor's software acceptance tests. Immediately following the independent testing the software program was functionally demonstrated for and accepted by the user. The importance of the overall

system schedule, however, forced the independent tester to test earlier baselines. This early testing, hopefully, would insure successful completion of the independent testing during the formally scheduled period.

Another area of concern was the criteria against which the independent tester evaluated the systems performance. In general, they used the same system specifications levied on the prime contractor. They worked closely with the user to determine if additional testing was required to demonstrate operational performance. The user in this case also relied solely on the system specifications. This was later changed by the SPO by adding a requirement to determine system limitations. The addition of this requirement relates to the statement contained in the OT&E paragraph on page 10 " . . . and to reevaluate the system to insure that it continues to meet operational needs and retains its effectiveness in a new environment or against a new threat."

This section provided an overview of tasks assigned to an Independent Software Validation Agency and examined several problems encountered during implementation. The next section will concentrate on the value of an independent tester and provide an updated tasking list for future acquisitions.

SECTION IV

TASKING INDEPENDENT TESTING AGENCIES

The System Development Corporation provided important and meaningful support to the Defense Support Program. The advantages of employing an independent tester far outweighed the advantages of relying totally on the prime contractor. The comparison of these advantages are similar to the comparison made internally when a software contractor is considering his testing philosophy. Table 2 and 3 (8:5-11)lists the advantages versus the disadvantages of testing with an independent source or a single source (develops the software) respectively.

TABLE 2

INDEPENDENT SOURCE

Advantages	Disadvantages
o Greater comprehensiveness of testing	o Cost of getting up to speed
o Utilization of a greater number of cost effective test techniques	o Greater overall cost
o More extensive testing	o Time consuming
o Elimination of subjectivity	o Inherent communications problems
o Independent verifier has no fear of reprimand for poor implementation	o Inclusion of unrealistic test and test data
o Better familiarity with the use of automated verification tools	
o Increased product confidence	

TABLE 3

SINGLE SOURCE

Advantages	Disadvantages
o Lower Cost	o Subjectivity (closed mind)
o Minimal communications	o Too involved with specific areas, usually not sufficiently testing all subroutine interfaces
o Debugging is simplified when a problem is encountered	o Verification is often regarded as a low priority item to be performed as time permits
o Avoidance of unrealistic or irrelevant test cases	o Lack of motivation and loss of interest
	o Single point of responsibility

The choice of either using "black box" or "cognizant testing" still must be made before an effective testing philosophy can be implemented.

The Air Force independent test agency (AFTEC) has to make the same basic choices before developing their test plans. To accomplish their testing they must also either develop an internal test capability or contract for the testing effort. Unless the user has developed an extensive capability in the

particular software development the user most likely will not be able to provide sufficient support to AFTEC. Keeping this in mind an updated and more realistic set of tasks has been developed for consideration by AFTEC. By excluding the first item the tasks could equally be considered as tasks for any Independent Software Validation Agency. The tasks are listed as follows:

1. Select a contractor to provide the software test capability.
2. The "cognizant testing" philosophy should be used. Implies the application of combined DT&E and OT&E where applicable.
3. Prepare an independent test plan by closely working with the developer and the user (Include DT&E test phases that will satisfy evaluation criterion).
4. Test the system against two criterions; a) system specifications b) operational requirements (design and current).
5. Include testing which allows maximum user interface with software program operations.
6. Maintain a fixed system software baseline. Apply this task following developmental command's acceptance from prime contractor.
7. Make all recommendations and criticisms of operational effectiveness against the baseline established in 6.
8. Prepare and maintain system test reports. Used

primarily for making recommendations on suitability for operational use but also important to use as basic building block for follow-on OT&E.

The key point in this list is the selection of the "cognizant testing" philosophy. Testing has already been identified as a costly (50 per cent) part of software development. The application of an independent test agency in the development of software as late as the OT&E would only lead to redundant tests and increase cost of the overall test program. "Cognizant testing" would reduce this redundancy by combining the developmental command's testing requirements with the independent tester. An additional benefit will also be incurred by allowing the independent agency to be involved early in the acquisition cycle. Of course, they would maintain their independent channel for reporting as well as their own objectivity.

Every effort should be made by the independent test agency to stay clear of the "chasing" the prime contractor's baseline. Selected functions of the software would be evaluated as a module and later tested in the system tests. These system tests would be conducted during Test (D) as shown in Figure 3.

This section has evaluated the advantages and disadvantages of independent testing. Also an updated and more realistic set of tasks were provided for consideration by AFTEC or by an Independent Software Validation Agency. The report will conclude

with recommendations for future acquisitions of Air Force weapon systems.

SECTION V

SUMMARY

The report has reviewed some of the documentation relating to Independent Software testing. The author has gained a better insight into the problems associated with the application of an Independent Software Validation Agency. The use of an actual software contract to illustrate the application of independent testing agencies supports the previous statement that the Air Force Systems Command has indeed been concerned about independent software validation. This example also provides a basic model for future consideration by the independent Air Force test agency (AFTEC).

Conclusions

Testing is an important and costly portion of software development and every effort should be made to reduce the cost of testing. This should not be done at the cost of system performance degradation but be carefully planned by all agencies involved. The use of an independent test agency will provide the critics of military testing an independent and objective evaluation of major weapon systems. Software development requires early involvement by all agencies. Waiting until the system is "delivered" increases the potential for costly and redundant testing.

Recommendations

The independent test agency selected to evaluate a particular software development should consider the list of tasks provided in this report.

The important points are to:

1. Get involved early in the program.
2. Maintain independence and objectivity.
3. Use all resources available to develop a concise and complete test plan.
4. Establish a baseline on which to evaluate the system.
5. Not expect the system to be 100 per cent successful in meeting the current or projected threat at the time of testing. (Dynamic Environment)
6. Provide suggested system improvements to meet the above threats.

Additional studies relating to Independent Software Validation Agencies should be conducted to insure that the software segments of major weapon systems are in fact being properly evaluated at reasonable and realistic cost.

BIBLIOGRAPHY

1. Bartlett, J. C., et al., Software Validation Study, Hq. SAMSO (AFSC), Los Angeles, Ca., SAMSO TR-73-105, March 1973.

2. Brosius, D. Bruce, et al., Software Validation Study, Final Report, HQ. SAMSO TR-76-99, February 1973.

3. Capps, Larry R., Software Management and The Testing of Weapon Systems That Contain an Embedded Computer System, Defense Systems Management School, Fort Belvoir, Va. Capps 75-2, November 1975.

4. DeRoze, Barry C., "An Introspective Analysis of DOD Weapon System Software Management", Defense Management Journal, V. 11, No. 4, October 1975.

5. Department of Defense Directive 5000.3, Test and Evaluation, May 20, 1975.

6. Dzezner, Stephen M., et al., The Computer Resource Management Study: Executive Summary, Rand Corporation, Santa Monica, CA., R-1855-PR, September, 1975.

7. Fisher, David A., "Programming Language Commodity in the Department of Defense", Defense Management Journal, V. 11, No. 4, October 1975.

8. Gibson, C. G., et al., Verification Guidelines, TRW Systems Group, Redondo Beach, CA., TRW-SS-71-04, August, 1971.

9. McCarthy, Rita, "Applying the Technique of Configuration Management to Software," Defense Management Journal, V. 11, No. 4., October 1975.

10. Miller, E. F., Jr., Methodology for Comprehensive Software Testing, General Research Corporation, Griffiss AFB, New York, RADC-TR-75-161, June 1975.

11. Pokorney, J. L. Captain, et al., A Systems Approach to Computer Programs, HQ. ESD (AFSC), Hanscom Field, Bedford, Mass., ESD-TR-67-205, February 1967.

12. Royce, Winston W., Managing The Development of Large Software Systems: Concepts and Techniques, TRW Systems Group, Redondo Beach, CA., TRW-SS-70-01, August, 1970.

13. Schwartz, Jules I., Analyzing Large Scale System Development, Report on a Conference Sponsored by The NATO Science Committee on Software Engineering Techniques, April 1970.

14. Software Project Manual, Air Force ADP, Draft June 1975.

15. Statement of Work, Defense Support Program, Software System Contractor, HQ SAMSO (AFSC) Los Angeles, CA., RFQ F04701-71-Q-0350, 13 October 1971.

16. Trainor, W. L., et al., Computer Program Testing, Technical Requirements and Standards Office, HQ. ESD (AFSC), Hanscom Field, Bedford, Mass., November 1971.

17. Vander Noot, T. J., "System Testing a Taboo Subject?", Selected Computer Articles, Department of Defense Computer Institute, July 1972.

18. Willmorth, N. E., Software Milestone Measurement Study, Naval Electronics Laboratory Center, San Diego, CA., NELC/TD 285, November 7, 1973.